REMARKS

This paper is responsive to a non-final Office action dated March 26, 2004. Claims 1-23 were examined. Claims 1 – 15 and 17 – 23 were rejected. The Examiner has indicated that claim 16 includes allowable subject matter, and Applicant appreciates such indication. Claim 1 has been amended to correct a grammar mistake, and not to overcome any prior art or for any other reason substantially related to patentability. Applicant respectfully traverses all rejections.

ExaminerTelephone Interview Summary

Applicant thanks the examiner for taking the time for the examiner interview on May 4, 2004. During the interview, the undersigned pointed out distinctions between the art relied upon and the pending claims. Those distinctions are detailed herein. No agreement was reached with respect to allowability of the claims.

Rejections in the Office Action

The Office relies upon a number of references in rejecting Applicant's claims. These references disclose some atomic operations on concurrent queues. However, the Office relies merely on disclosure of an atomic operation within the context of a concurrent queue to reject Applicant's claims, and overlooks the actual language of Applicant's claims, including, if applicable, atomic operations. Applicant respectfully submits that merely disclosing an atomic operation within the context of a concurrent queue does not support rejections of Applicant's claims. In some instances, the Office quotes a sentence from a reference to characterize the disclosure of the reference, and fails to take note of subsequent statements and sections of the reference that contradict the characterization. The Office also extracts unrelated statements from different sections of a reference and combines these statements to advance assertions that are not supported by the reference.

Rejections under 35 U.S.C. §102

35 U.S.C. §102(b)

The Office Action rejects claims 1, 3 – 5, and 11 under 35 U.S.C. §102(b) as being unpatentable over "Non-Blocking Algorithms for Concurrent Data Structures" by Prakash, et al.

("Prakash"). The Office Action also rejects claims 12 – 15 and 18 – 19 under 35 U.S.C. §102(b) as being unpatentable over "A Nonblocking Algorithm for Shared Queues Using Compare-and-Swap" by Prakash, et al. ("Prakash 2"). The Office Action also rejects claims 20 – 23 under 35 U.S.C. §102(b) as being unpatentable over "Simple, Fast, and Practical Non-Blocking and Blocking Concurrent Queue Algorithms" by Michael, et al. ("Michael"). Applicant respectfully traverses all of these rejections.

Prakash

The Office rejects independent claim 1 based on a statement in Prakash that an enqueuing stream and a dequeuing stream can concurrently access a queue, when there are two or more elements in the queue. The Office then extracts a statement from a different section of Prakash which merely discloses checking state of the queue as part of an enqueue operation. The Office uses these two statements to support an erroneous assertion that Prakash discloses a dequeue and an enqueue stream accessing the queue throughout a complete range of valid states, including a boundary condition state. Within the statement relied upon by the Office, it specifically states that concurrent access to the queue by an enqueue and a dequeue stream is limited to *when there are two or more elements in the queue.* In addition, Prakash states at page 11, "However, enqueuers and dequeurs can proceed in parallel if there is more than one element in the queue. Otherwise, they must proceed serially... Also, we need to put guards in the data structure to combine these two streams in a single stream when there is only a single element in the queue."

As reflected by these statements within Prakash, Prakash's technique is simply not "non-blocking with respect to any other execution of the access operations throughout a complete range of valid states" as recited in claim 1. Clearly, Prakash's technique is not non-blocking for either of the following valid states: an empty state or a single element state. Hence, Prakash does not anticipate claim 1, which includes the following:

> wherein execution of any one of the access operations is **non-blocking with respect to any other execution of the access operations throughout a complete range of valid states, including one or more boundary condition states**....

Furthermore, Prakash's technique does not disclose opposing-end access operations that are disjoint, as evidenced by the use of a Current_op variable utilized to track commencement

and completion of an access operation (see Figure2 of Prakash). Hence, Prakash does not disclose or suggest "at least for those of the valid states other than the one or more boundary condition states, opposing-end ones of the access operations are disjoint" as recited in claim 1.

Prakash 2

The Office rejects independent claim 12 based on an assumption about Prakash 2 that does not have support within Prakash 2. The Office simply assumes that Prakash 2 discloses atomically updating a node, a deleted node indication, and a list-end identifier, as found in claim 12. However, there is no such disclosure within Prakash 2. In fact, the dequeue procedure illustrated in Figure 6 of Prakash 2 marks an object to be dequeued with a first *compare&swap double* (CSDBL) operation, sets the tail to null with a second CSDBL operation, and then shifts the head to null with a third CSDBL operation. Three separate atomic operations are performed in Prakash 2 as part of its dequeue procedure. Prakash 2 does not suggest or disclose "executing as part of a pop operation, **an atomic update of a list node and both a deleted node indication and list-end identifier corresponding thereto**" as in claim 12, or "executing as part of a delete operation, **an atomic update of a deleted node indication and at least one list-end identifier corresponding thereto**" as in claim 13. Again, Prakash discloses performing each update separately and not as an atomic update.

Claim 14 recites "responsive to the deleted node indication, excising a marked node from the list by atomically updating opposing direction pointers impinging thereon and the deleted node indication thereto." Prakash 2 does not disclose atomically updating opposing direction pointers to excise a node, as presumed by the Office. Prakash 2 discloses setting a single pointer to NULL in Figure 6. Prakash 2 is void of any disclosure of atomically updating opposing direction pointers to excise a marked node.

Claim 15 recites "deleting the marked element from the list at least before completion of a same end push or pop operation." Prakash 2 fails to disclose or suggest the subject matter of claim 15. The Office refers to the delink operation of the dequeue procedure in Figure 6 of Prakash 2. Figure 6 explicitly illustrates that an object will not be delinked until after completion of an unfinished dequeue or enqueue procedure.

Claims 18 and 19 recite encoding the deleted node indication differently. Claim 18 recites encoding the deleted node indication integral with an end-node identifying pointer and claim 19 recites encoding the deleted node indication as a dummy node. Prakash 2 does not disclose either of these methods of encoding the deleted node indication. Prakash 2 clearly illustrates encoding the deleted node indication integral with the corresponding node, and not as a dummy node and not integral with an end-identifying node.

Michael

Once again, the Office makes presumptions as to the contents of a reference. When utilizing Michael to reject claims 20 – 23, the Office refers to both non-blocking and blocking schemes disclosed in Michael, even though the claims specifically recite "non-blocking access". The Office presumes that the enqueue and dequeue procedures disclosed by Michael atomically update an end-identifier of a linked list and a corresponding node of the linked list. However, neither the enqueue operation nor the dequeue operation disclosed by Michael supports the Office's presumptions. The enqueue operation disclosed by Michael updates the tail pointer, but does not update a node. The dequeue operation disclosed by Michael updates the head pointer, and later frees an old dummy node. Michael does not disclose or suggest "an atomic operation to **atomically update one of the end identifiers and a node of the linked-list corresponding thereto**, wherein for opposing end instances, the atomic updates are disjoint for at least all non-empty states of the concurrent shared object" as in claim 20. Michael does not disclose or suggest opposing end instances and especially does not disclose or suggest disjoint atomic updates of opposing end instances.

In rejecting independent claim 23, the Office simply refers to its rejection of claim 20. Yet, claim 23 recites the following:

> means for coordinating competing pop operations, the coordinating means employing in each instance thereof, an atomic operation to disambiguate a retry state and a boundary condition state of the concurrent shared object based on then-current contents of one, but not both, of the first- and second-end identifier stores and an element of the concurrent shared object corresponding thereto.

There is nothing in Michael that discloses claim 23 and the Office has not indicated any such disclosure within Michael. The rejection relies on Michael's dequeue procedure that discloses reading a value of a node, and then attempting to swing Head to the next node with a compare&swap (CAS) operation. Michael does not disclose coordinating competing pop operations and disambiguating a retry state and a boundary condition state with an atomic operation.

35 U.S.C. §102(a)

The Office Action rejects claims 1, 2, 6, and 7 under 35 U.S.C. §102(e) as being unpatentable over "Non-blocking Synchronization and System Design" by Michael Barry Greenwald ("Greenwald"). Applicant respectfully notes that Greenwald (a non-patent reference) does not meet the requirements of 35 U.S.C. §102(e). Greenwald is potentially a reference under 35 U.S.C. §102(a), and Applicant responds to the rejection accordingly, although Applicant reserves the right to swear behind the date of publication of Greenwald. Applicant respectfully traverses all rejections under Greenwald.

Greenwald

In addition to rejecting claim 1 with Prakash, the Office rejects claim 1 with Greenwald, but Greenwald also fails to achieve Applicant's claim 1. Greenwald assumes an array of unbounded size, and does not deal with issues such as detection of when the dequeue is empty or full. Greenwald relies on augmentation of the elements of a dequeue and ignores the practical limitations of a data structure, which includes the size of the data structure. In particular, Greenwald's list based design has limited in applicability at least because it cuts the index range of a list to half a memory word, and Greenwald's list based design prevents concurrent access to the two ends of a list. Furthermore, Greenwald's array based design assumes an array of unbounded size, and does not deal with classical array-based issues, such as detection of boundary conditions. Neither the list-based nor the array-based designs proposed by Greenwald disclose or suggest 1) access operations that are non-blocking throughout a complete range of valid states nor 2) opposing-end access operations that are disjoint. Accordingly, Greenwald does not disclose of suggest "wherein execution of any one of the access operations is non-blocking with respect to any other execution of the access operations throughout a complete range of valid states, including one or more boundary condition states, and wherein, at least for

those of the valid states other than the one or more boundary condition states, opposing-end ones of the access operations are disjoint" as recited in claim 1.

Rejections under 35 U.S.C. §103(a)

The Office Action rejects claims 9 and 10 under 35 U.S.C. §103(a) as being unpatentable over Prakash in view of Prakash 2. The Office Action also rejects claim 17 under 35 U.S.C. §103(a) as being unpatentable over Prakash. The Office Action also rejects claim 8 under 35 U.S.C. §103(a) as being unpatentable over Greenwald. Applicant respectfully traverses all of these rejections.

As stated in the argument with respect to claims 12 and 13, neither Prakash nor Prakash 2 disclose or teach "an atomic update of a deleted node indication and...list-end identifier corresponding thereto" as recited in claims 9 and 10.

With regard to claim 17, Prakash 2 does not disclose or suggest a DCAS operation as recited in claim 17. Prakash 2 recites a compare&swap double operation (CSDBL). The semantics of these two operations are different. The DCAS operation is well-known in the art, but exemplary code for a DCAS can be found at page 8 of Applicant's specification. The Examiner states that it would be obvious to apply Prakash 2 to doubly-linked list. However, the Examiner does not provide any support for such an assertion beyond the statement within the Office Action. Applicant respectfully requests identification within Prakash 2 of support for the assertion.

With regard to claim 8, Greenwald discloses atomic updating of opposing end-indices of a queue in Figures D.19 and D.20. In contrast to the mistaken characterization by the Office, Greenwald does not disclose or suggest atomically updating an opposing end index and a corresponding array element. More specifically, Greenwald does not disclose or suggest "an atomic update of a respective one of the opposing-end indices and of an array element corresponding thereto" as recited in claim 8.

For at least the reasons stated above, Applicant respectfully submits that all of Applicant's independent claims are patentable over Prakash, Prakash 2, Michael, and Greenwald, standing alone or in combination, and are therefore allowable. Furthermore, the dependent

claims depend on corresponding ones of the above allowable independent claims and are allowable at least for the reasons explained above.

CONCLUSION

In summary, claims 1-23 are in the case. All claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that, on the date shown below, this correspondence is being

☐ deposited with the US Postal Service with sufficient postage as first class mail, in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

☒ facsimile transmitted to the US Patent and Trademark Office.

Steven R. Gilliam — Jun-28-2004 (Date)

EXPRESS MAIL LABEL: _____

Respectfully submitted,

Steven R. Gilliam, Reg. No. 51,734
Attorney for Applicant(s)
(512) 338-6320
(512) 338-6301 (fax)